
Scalable Matrix-valued Kernel Learning and High-dimensional Nonlinear Causal Inference

Vikas Sindhwani
IBM Research
Yorktown Heights, NY 10598
vsindhw@us.ibm.com

Aur lie C. Lozano
IBM Research
Yorktown Heights, NY 10598
aclozano@us.ibm.com

Ha Quang Minh
Istituto Italiano di Tecnologia
Genoa, 16163, Italy
minh.haquant@iit.it

Version as of June 1st, 2012

Abstract

We propose a general matrix-valued multiple kernel learning framework for high-dimensional nonlinear multivariate regression problems. This framework allows a broad class of mixed norm regularizers, including those that induce sparsity, to be imposed on a dictionary of vector-valued Reproducing Kernel Hilbert Spaces [19]. We develop a highly scalable and eigendecomposition-free Block coordinate descent procedure that orchestrates two inexact solvers: a Conjugate Gradient (CG) based Sylvester equation solver for solving vector-valued Regularized Least Squares (RLS) problems, and a specialized Sparse approximate SDP solver [15] for learning output kernels. As an application of our framework, we show how high-dimensional causal inference tasks can be naturally cast as sparse function estimation problems within our framework, leading to novel nonlinear extensions of Grouped Graphical Granger Causality techniques. The algorithmic developments and extensive empirical studies are complemented by theoretical analyses in terms of Rademacher generalization bounds.

1 Introduction

This paper is at the intersection of three distinct but symbiotic themes in machine learning and statistics: (a) non-parametric multivariate regression and structured output learning, (b) sparse learning for high-dimensional settings [7], and (c) multiple time series analysis [17] and associated temporal causal modeling problems [21, 16, 25]. We begin by considering the general problem of estimating an unknown non-linear function $f : \mathcal{X} \mapsto \mathcal{Y}$ from labeled examples, where the output space \mathcal{Y} has a Hilbert space structure. When \mathcal{Y} is finite dimensional, the problem is akin to multivariate regression. The presence of possible correlations amongst outputs naturally motivates more effective learning algorithms that attempt to learn all coordinates jointly instead of treating them independently. Such problems can be formulated as regularized risk minimization over a \mathcal{Y} -valued hypothesis space of functions for which a general Reproducing Kernel Hilbert Space (RKHS) framework has been recently brought to the attention of machine learning community by [19]. However, the Kernel function in this setting becomes matrix-valued whose choice turns into a challenging model selection problem – certainly much more exaggerated than in the scalar case where Gaussian or Polynomial kernels are a default choice that require only a few hyperparameters to be tuned. Compounded further by the computational complexity of solving the resulting optimization problems involving large dense matrices, vector-valued extensions of kernel methods are arguably yet to find widespread application, despite the fact that their theory can be traced as far back as the work of Laurent Schwarz in 1964 [24]. Our contributions in this paper are as follows:

- We first seek a scalable resolution of the kernel learning problem in vector-valued settings. Towards this end, we propose a general framework for function estimation over a dictionary of vector-valued RKHSs where a broad family of variationally defined regularizers, including sparsity inducing norms, serve to optimally combine a collection of matrix-valued kernels. As such

our framework may be viewed as non-parameteric multivariate extension of Group Lasso and related sparse learning models [7].

- An overview on various classes of matrix valued kernels is given in [1]. However, in this paper, we restrict attention to *separable kernels* due to their universality [8], conceptual simplicity and potential for scalability. Separable matrix-valued kernels are composed of a *scalar input kernel function* and an *output kernel matrix* (to be formally defined later). We focus on vector-valued Regularized Least Squares (RLS) algorithms which lead to *Sylvester matrix equations* of a specific form that can be solved in cubic time using appropriate eigendecompositions. These solvers are much more efficient than RLS models with general matrix-valued kernels, but nonetheless become a significant computational bottleneck [11] when invoked repeatedly in the larger context of a kernel learning algorithm. To exploit warm starts and additional structure in the problem, we develop a Conjugate Gradient (CG) based Sylvester equations solver. Our block coordinate descent approach jointly optimizes input scalar kernel combinations and the output kernel matrix, where for the latter we specialize the Sparse SDP solver of [15] for inexact but efficient optimization over the cone of positive semi-definite matrices with bounded trace.
- Empirical results confirm the value of joint optimization of input and output kernels. The use of inexact solvers greatly improves the scalability of our algorithms.
- We provide bounds on Rademacher Complexity for the hypothesis sets considered by our algorithm. In particular, this extends the results of [10] to the vector-valued case.
- We then turn to multiple time series analysis. We propose to replace classic linear Vector Autoregression (VAR) models traditionally used in time series analysis ([17]), with sparse nonlinear vector-valued RLS models. In particular, we consider the problem of forecasting the evolution of a large collection of high-dimensional time series variables, potentially given a small set of observations. Various variable groupings, e.g., lagged values of temporal variables, induce a candidate pool of kernels from which an optimal small subset is selected by our kernel learning procedures. The resulting sparse model can then be naturally endowed with a Graphical Granger Causality interpretation. This provides a link between nonparameteric notions of sparsity developed in the multiple kernel learning literature, and operational notions of Causality developed by Clive Granger [14]. We illustrate the power of our approach on computational biology problems.

2 Vector-valued RLS & Separable Matrix-valued Kernels

We begin by providing a brief background and setting up some notation. Let us focus on the Regularized Least Squares (RLS) framework for finite dimensional multivariate regression settings. Given labeled examples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^l$, $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$, $\mathbf{y}_i \in \mathcal{Y} \subset \mathbb{R}^n$ the vector-valued RLS solves the following problem,

$$\arg \min_{f \in \mathcal{H}_{\vec{k}}} \frac{1}{l} \sum_{i=1}^l \|f(\mathbf{x}_i) - \mathbf{y}_i\|_2^2 + \lambda \|f\|_{\mathcal{H}_{\vec{k}}}^2 \quad (1)$$

where $\mathcal{H}_{\vec{k}}$ is a vector-valued RKHS generated by the kernel function \vec{k} , and $\lambda > 0$ is the regularization parameter. In the vector-valued setting, \vec{k} is a matrix-valued function, i.e., for any pair of inputs \mathbf{x}, \mathbf{z} , $\vec{k}(\mathbf{x}, \mathbf{z})$ is an n -by- n matrix, or more generally speaking, an input-dependent linear operator on the output space. The kernel function is positive in the sense that for any finite set of m input-output pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, the following holds: $\sum_{i,j=1}^m \mathbf{y}_i^T \vec{k}(\mathbf{x}_i, \mathbf{x}_j) \mathbf{y}_j \geq 0$. A generalized representer theorem says that the optimal solution has the form, $f(\cdot) = \sum_{i=1}^l \vec{k}(\mathbf{x}_i, \cdot) \alpha_i$ where the coefficients α_i are n -dimensional vectors. For RLS, these coefficient vectors can be obtained solving a dense linear system, of the familiar form, $(\vec{\mathbf{K}} + \lambda \mathbf{I}_{nl}) \text{vec}(\mathbf{C}^T) = \text{vec}(\mathbf{Y}^T)$, where $\mathbf{C} = [\alpha_1 \dots \alpha_l] \in \mathbb{R}^{l \times n}$ assembles the coefficient vectors into a matrix; the *vec* operator stacks columns of its argument matrix into a long column vector; $\vec{\mathbf{K}}$ is a large $nl \times nl$ -sized Gram matrix comprising of the blocks $\vec{k}(\mathbf{x}_i, \mathbf{x}_j)$, for $i, j = 1 \dots l$, and \mathbf{I}_{nl} denotes the identity matrix of compatible size. It is easy to see that for $n = 1$, the above developments exactly collapse to familiar concepts for scalar RLS (also known as Kernel Ridge Regression). In general though, the above linear system requires $O((nl)^3)$ time to be solved using standard dense numerical linear algebra, which is clearly prohibitive. However, for a family of *separable matrix-valued kernels* defined below, the

computational cost can be improved to $O(n^3 + l^3)$ which, though still costly, is comparable to scalar RLS when l is much larger than n .

Separable Matrix Valued Kernel and its Gram matrix: Let k be a scalar kernel function on the input space \mathcal{X} and \mathbf{K} represent its gram matrix on a finite sample. Let \mathbf{L} be an $n \times n$ positive semi-definite output kernel matrix. Then, the function $\vec{k}(\mathbf{x}, \mathbf{z}) = k(\mathbf{x}, \mathbf{z})\mathbf{L}$ is positive and hence defines a matrix valued kernel. The gram matrix of this kernel is $\vec{\mathbf{K}} = \mathbf{K} \otimes \mathbf{L}$ where \otimes denotes Kronecker product.

For separable kernels, the corresponding RLS dense linear system (Eqn 2 below) can be reorganized into a Sylvester equation (Eqn 3 below):

$$(\mathbf{K} \otimes \mathbf{L} + \lambda \mathbf{I}_{nl}) \text{vec}(\mathbf{C}^T) = \text{vec}(\mathbf{Y}^T) \quad (2)$$

$$\mathbf{KCL} + \lambda \mathbf{C} = \mathbf{Y} \quad (3)$$

Sylvester solvers are more efficient than applying a direct dense linear solver for Eqn 2. The classical Bartel-Stewart and Hessenberg-Schur¹ methods are usually used for solving Sylvester equations. They are similar in flavor to an eigendecomposition approach we describe next for completeness, though they take fewer floating point operations at the same cubic order of complexity.

Eigen-decomposition based Sylvester Solver: Let $\mathbf{K} = \mathbf{TMT}^T$ and $\mathbf{L} = \mathbf{SNS}^T$ denote the eigen-decompositions of \mathbf{K} and \mathbf{L} respectively where \mathbf{T}, \mathbf{S} are orthonormal matrices, and let the eigenvalues be denoted by $\mathbf{M} = \text{diag}(\sigma_1 \dots \sigma_l), \mathbf{N} = \text{diag}(\rho_1 \dots \rho_n)$. Then the solution to the matrix equation $\mathbf{KCL} + \lambda \mathbf{C} = \mathbf{Y}$ always exists when $\lambda > 0$ and is given by $\mathbf{C} = \mathbf{T}\tilde{\mathbf{X}}\mathbf{S}$ where $\tilde{\mathbf{X}}_{ij} = \frac{(\mathbf{T}^T \mathbf{Y} \mathbf{S})_{ij}}{\sigma_i \rho_j + \lambda}$.

Output Kernel Learning: We will use the shorthand $\vec{k} = k\mathbf{L}$ to represent the implied separable kernel and correspondingly denote its RKHS by $\mathcal{H}_{k\mathbf{L}}$. In recent work [11] develop an elegant extension of the vector-valued RLS problem, Eqn. 1, to jointly learn both $f \in \mathcal{H}_{k\mathbf{L}}$ and \mathbf{L} . In finite dimensional language, a function $f(\mathbf{x}) = \mathbf{L}\mathbf{C}\mathbf{k}_{\mathbf{x}} \in \mathcal{H}_{k\mathbf{L}}$ is estimated by solving the following problem,

$$\arg \min_{\mathbf{C} \in \mathbb{R}^{l \times n}, \mathbf{L} \in \mathcal{S}_+^n} \frac{1}{l} \|\mathbf{KCL} - \mathbf{Y}\|_{fro}^2 + \lambda \text{trace}(\mathbf{C}^T \mathbf{KCL}) + \rho \|\mathbf{L}\|_{fro}^2 \quad (4)$$

where \mathcal{S}_+^n denotes the cone of positive semi-definite matrices. It is shown that the objective function is *invex*, i.e., its stationary points are globally optimal. [11] proposed a block coordinate descent where for fixed \mathbf{L} , \mathbf{C} is obtained by solving Eqn. 3 using an Eigendecomposition-based solver. Under the assumption that \mathbf{C} exactly satisfies Eqn. 3, the resulting update for \mathbf{L} is then shown to automatically satisfy the constraint that $\mathbf{L} \in \mathcal{S}_+^n$. However, [11] remark that experiments on their largest dataset took roughly a day to complete on a standard desktop and that the “limiting factor was the solution of the Sylvester equation”.

3 Learning over a Vector-valued RKHS Dictionary

We seek a fuller resolution of the separable kernel learning problem for vector-valued RLS problems, which is eigendecomposition-free and more scalable. In this section, we expand Eqn. 4 to simultaneously learn both input and output kernels over a predefined dictionary, and develop optimization algorithms based on approximate solvers that execute cheap iterations. Consider a **dictionary of separable matrix valued kernels** sharing the same output kernel: $\mathcal{D}_{\mathbf{L}} = \{k_1\mathbf{L}, \dots, k_m\mathbf{L}\}$. It is possible for some of the scalar kernels to arise from specific groups of features, i.e. $k_i(\mathbf{x}, \mathbf{z}) = g_i(P_i\mathbf{x}, P_i\mathbf{z})$ where P_i is a projection from \mathcal{X} to a group of features $\mathcal{X}_i \subset \mathcal{X}$, and $g_i : \mathcal{X}_i \times \mathcal{X}_i \mapsto \mathbb{R}$ is a scalar kernel on \mathcal{X}_i . Let $\mathcal{H}(\mathcal{D}_{\mathbf{L}})$ denote the sum space of functions $\mathcal{H}(\mathcal{D}_{\mathbf{L}}) = \{f = \sum_j f_j | f_j \in \mathcal{H}_{k_j\mathbf{L}}, j = 1 \dots m\}$ and equip this space with the following l_p norms: $\|f\|_{l_p(\mathcal{H}(\mathcal{D}_{\mathbf{L}}))} = \inf_{f = \sum_j f_j} \left\| \left(\|f_1\|_{\mathcal{H}_{k_1\mathbf{L}}}, \dots, \|f_m\|_{\mathcal{H}_{k_m\mathbf{L}}} \right) \right\|_p$. Note that $\|f\|_{l_1(\mathcal{H}(\mathcal{D}_{\mathbf{L}}))}$, being the l_1 norm of the vector of norms in individual RKHSs, imposes a functional notion of sparsity on the

¹Implemented in SLICOT and available in MATLAB via dlyap inbuilt function.

vector-valued function f . We now consider objective functions of the form,

$$\arg \min_{f \in \mathcal{H}(\mathcal{D}_L), \mathbf{L} \in \mathcal{S}_+^n(\tau)} \frac{1}{l} \sum_{i=1}^l \|f(\mathbf{x}_i) - \mathbf{y}_i\|_2^2 + \lambda \Omega(f) + \Gamma(\mathbf{L}) \quad (5)$$

where \mathbf{L} is constrained to belong to the *spectahedron* with bounded trace: $\mathcal{S}_+^n(\tau) = \{\mathbf{X} \in \mathcal{S}_+^n | \text{trace}(\mathbf{X}) \leq \tau\}$ where \mathcal{S}_+^n denotes the cone of symmetric positive semi-definite matrices, Γ is a convex differentiable penalty function over \mathbf{L} (e.g., $\rho\|\mathbf{L}\|_{fro}^2$), and Ω is a regularizer whose canonical choice is $\Omega(f) = \|f\|_{l_p(\mathcal{H}(\mathcal{D}_L))}$. Our algorithms work for a broad choice of regularizers that admit quadratic variational representation of the form

$$\Omega(f) = \arg \min_{\boldsymbol{\eta} \in \mathbb{R}_+^m} \sum_{i=1}^m \frac{\|f_j\|_{\mathcal{H}_{k_i}}^2}{\eta_i} + \omega(\boldsymbol{\eta}) \quad (6)$$

for an appropriate function $\omega : \mathbb{R}_+^m \mapsto \mathbb{R}$. We rationalize this framework as follows: Penalty functions of the form above define a broad family of structured sparsity-inducing norms that have extensively been used in the multiple kernel learning and sparse modeling literature [4, 26]. They allow complex non-differentiable norms to be related back to weighted RKHS norms, and optimizing $\boldsymbol{\eta}$ in many cases infact admits closed form expressions. Optimizing \mathbf{L} over the Spectahedron allows us to develop a specialized version of the approximate Sparse SDP solver [15] whose iterations involve the computation of only a single extremal eigenvector of the (partial) gradient at the current iterate – this involves relatively cheap operations followed by quick rank-one updates. Furthermore, by bounding the trace of \mathbf{L} , we show below that a Conjugate Gradient (CG) based iterative Sylvester solver for Eqn. 2 would always be invoked on well-conditioned instances and hence show rapid numerical convergence (particularly also with warm starts). The trace parameter τ also naturally appears in our Rademacher generalization analyses.

First we observe that a basic result concerning sums of scalar RKHSs also holds unsurprisingly for the vector-valued case. The proof given in Appendix A follows Section 6 of [3] replacing scalar concepts with corresponding notions from the theory of vector-valued RKHSs [19].

Proposition 1. *Given a collection of operator-valued reproducing kernels $\vec{k}_1 \dots \vec{k}_m$ and positive scalars $\eta_j > 0, j = 1 \dots m$, the function $\vec{k}_\boldsymbol{\eta} = \sum_{i=1}^m \eta_i \vec{k}_i$ is the reproducing kernel of the sum space $\mathcal{H} = \{f : \mathcal{X} \mapsto \mathcal{Y} | f(\mathbf{x}) = \sum_{j=1}^m \eta_j f_j(\mathbf{x}), f_j \in \mathcal{H}_{\vec{k}_j}\}$ with the norm given by $\|f\|_{\mathcal{H}_{\vec{k}_\boldsymbol{\eta}}}^2 = \arg \min_{f = \sum_{j=1}^m f_j, f_j \in \mathcal{H}_{\vec{k}_j}} \sum_{j=1}^m \frac{\|f_j\|^2}{\eta_j}$.*

This result combined with the variational representation of the the penalty function in Eqn. 6 allows us to reformulate Eqn. 5 in terms of a joint optimization problem over $\boldsymbol{\eta}, \mathbf{L}$ and $f \in \mathcal{H}_{k_\boldsymbol{\eta}}$, where we define the weighted scalar kernel $k_\boldsymbol{\eta} = \sum_j \eta_j k_j$. This formulation allows us to scale gracefully with respect to m , the number of kernels. Denote the Gram matrix of $k_\boldsymbol{\eta}$ on the labeled data as $\mathbf{K}_\boldsymbol{\eta}$, i.e., $\mathbf{K}_\boldsymbol{\eta} = \sum_{j=1}^p \eta_j \mathbf{K}_j$, where \mathbf{K}_j we denotes the gram matrices of the individual scalar kernel k_j . The finite dimensional version of the reformulated problem becomes,

$$\mathcal{O}(\boldsymbol{\eta}, \mathbf{C}, \mathbf{L}) = \frac{1}{l} \|\mathbf{K}_\boldsymbol{\eta} \mathbf{C} \mathbf{L} - \mathbf{Y}\|_{fro}^2 + \lambda \text{trace}(\mathbf{C}^T \mathbf{K}_\boldsymbol{\eta} \mathbf{C} \mathbf{L}) + \Gamma(\mathbf{L}) + \omega(\boldsymbol{\eta}) \quad (7)$$

over $\mathbf{C} \in \mathbb{R}^{n \times l}$, $\mathbf{L} \in \mathcal{S}_+^n(\tau)$ and $\boldsymbol{\eta} \in \mathbb{R}_+^m$. The auxillary function $\omega(\boldsymbol{\eta})$ depends on the specific form of $\Omega(f)$; for l_p norms it is the indicator function of the set $\eta_i \geq 0, \sum_{i=1}^m \eta_i^q \leq 1$ where $\frac{1}{p} + \frac{1}{q} = 1$. A natural strategy for such a problem is Block Coordinate Descent². At termination, the vector-valued function returned is $f^*(\mathbf{x}) = \mathbf{L} \mathbf{C} [k_\boldsymbol{\eta}(\mathbf{x}, \mathbf{x}_1) \dots k_\boldsymbol{\eta}(\mathbf{x}, \mathbf{x}_l)]^T$. We next describe each block minimization problem.

Conjugate Gradient Sylvester Solver: For fixed $\boldsymbol{\eta}, \mathbf{L}$, the optimal \mathbf{C} is given by the solution of the dense linear system of Eqn 2 or the Sylvester equation 3, with $\mathbf{K} = \mathbf{K}_\boldsymbol{\eta}$. General dense linear solvers have prohibitive $O(n^3 l^3)$ cost when invoked on Eqn. 2. The $O(n^3 + l^3)$ eigendecomposition-based Sylvester solver performs much better, but needs to be invoked repeatedly since \mathbf{L} as well as $\mathbf{K}_\boldsymbol{\eta}$ are changing across (outer) iterations. Instead, we apply a CG-based iterative solver for Eqn 2.

²Concerning BCD convergence in such a context, see remarks about smoothing operations in [4]

Despite the large size of the linear system, using CG infact has several quantifiable advantages due to the special Kronecker structure of Eqn. 2:

- A CG solver can exploit warm starts by initializing from previous $\boldsymbol{\eta}$, \mathbf{L} , and allow early termination at cheaper computational cost.
- The large $nl \times nl$ coefficient matrix in Eqn.2 never needs to be explicitly materialized. For any CG iterate $\mathbf{X}^{(i)}$, matrix-vector products can be efficiently computed since $(\mathbf{K}_\eta \otimes \mathbf{L} + \lambda \mathbf{I}_{nl}) \text{vec}(\mathbf{X}^{(i)T}) = \text{vec}(\mathbf{K}_\eta \mathbf{X}^{(i)} \mathbf{L} + \lambda \mathbf{X}^{(i)})$. CG can exploit additional low-rank or sparsity structure in \mathbf{K}_η and \mathbf{L} for fast matrix multiplication. When the base kernels are either (a) linear kernels derived from a small group of features, or (b) arise from randomized approximations, such as the random Fourier features for Gaussian Kernel [22], then $\mathbf{K}_\eta = \sum_{j=1}^m \eta_j \mathbf{Z}_j \mathbf{Z}_j^T$ where \mathbf{Z}_j has $d_j \ll l$ columns. In this case, \mathbf{K}_η need never be explicitly materialized and the cost of matrix multiplication can be further reduced.
- CG is expected to make rapid progress in a few iterations in the presence of a strong regularizer. This is because the coefficient matrix is expected to be well conditioned. Assume that $\phi = \max_{j \supset \mathbf{x} \in \mathcal{X}} k_j(\mathbf{x}, \mathbf{x})$ for all j , as in a dictionary of Gaussian Kernels where $\phi = 1$ and that $\sum_j \eta_j \leq 1$, which holds for l_1 norm. Then, the condition number (ratio of largest to smallest eigenvalue) of the matrix $(\mathbf{K}_\eta \otimes \mathbf{L} + \lambda \mathbf{I})$ can be bounded as $\kappa = \frac{\sigma_1 \rho_1 + \lambda}{\sigma_l \rho_n + \lambda} \leq \frac{\phi \tau}{\lambda} + 1$, where σ_1, ρ_1 (and σ_l, ρ_n) are the largest (smallest) eigenvalues of \mathbf{K} and \mathbf{L} respectively. Specializing standard CG convergence results [6] to our setting, we see that τ, λ control how fast CG will make progress: $\|\mathbf{C}^{(k)} - \mathbf{C}^*\|_{fro} \leq 2\sqrt{\kappa} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^k \|\mathbf{C}^{(0)} - \mathbf{C}^*\|_{fro}$ where $\mathbf{C}^{(k)}$ is the solution at step k , \mathbf{C}^* is the optimal solution (at current fixed $\boldsymbol{\eta}$ and \mathbf{L}) and $\mathbf{C}^{(0)}$ is the initial iterate.

Updates for $\boldsymbol{\eta}$: Existing results [4, 26, 20] from MKL literature can be adapted to our setting to get closed form update rules listed below for some choices of $\Omega(f)$. Let $\alpha_j = \hat{\eta}_j \sqrt{\text{trace}(\mathbf{C} \mathbf{K}_j \mathbf{C} \mathbf{L})}$ where $\hat{\eta}_j$ refers to previous value of η_j . For l_1 penalty, $\|f\|_{l_p(\mathcal{H}(\mathcal{D}_L))}^2$, the optimal $\eta_j = \alpha_j^{\frac{2}{r+1}} / \left(\sum_{j=1}^m \alpha_j^{\frac{2}{r+1}} \right)^r$ for $r = \frac{2}{2-p}$. For an elastic net style penalty, $(1 - \mu)\|f\|_{l_1(\mathcal{H}(\mathcal{D}_L))} + \mu\|f\|_{l_2(\mathcal{H}(\mathcal{D}_L))}^2$, the optimal $\eta_j = \frac{\alpha_j}{1 - \mu + \mu \alpha_j}$. Other choices are possible, e.g., see Table 1 in [26].

Spectahedron Solver: Here, we consider the \mathbf{L} optimization subproblem, which is: $\arg \min_{\mathbf{L} \in \mathcal{S}_+^n(\tau)} g(\mathbf{L}) = \frac{1}{T} \|\mathbf{A} \mathbf{L} - \mathbf{Y}\|_{fro}^2 + \lambda \text{trace}(\mathbf{B}^T \mathbf{L}) + \Gamma(\mathbf{L})$ where $\mathbf{A} = \mathbf{K}_\eta \mathbf{C}$, $\mathbf{B} = \mathbf{C}^T \mathbf{A}$. Hazan's Sparse SDP solver [15, 12] based on Frank-Wolfe algorithm [9], can be used for problems of the general form, $\mathbf{L}^* = \arg \min_{\mathbf{L} \in \mathcal{S}_+^n, \text{trace}(\mathbf{L})=1} g(\mathbf{L})$, where g is a convex, symmetric and differentiable function. In each iteration, it optimizes a linearization of the objective function around the current iterate \mathbf{L}_k , resulting in updates of the form, $\mathbf{L}_{k+1} = \mathbf{L}_k + \alpha_k (\mathbf{v}_k \mathbf{v}_k^T - \mathbf{L}_k)$ where $\mathbf{v}_k = \text{ApproxEV} \left(\nabla g(\mathbf{L}_k), \frac{C_g}{k^2} \right)$ and $\alpha_k = \min \left(1, \frac{2}{k} \right)$. Here, *ApproxEV* is an approximate eigensolver which when invoked on the gradient of g at the current iterate \mathbf{L}_k (a positive semi-definite matrix) computes the single eigenvector corresponding to the smallest eigenvalue, to a prespecified precision $C_g k^{-2}$, where C_g is a curvature constant upper bounded by the largest eigenvalue of the Hessian of g . Hazan's algorithm is appealing since each iteration itself tolerates approximations, the updates pump in rank-one terms, and it comes with the guarantee that after k steps, $g(\mathbf{L}_{k+1}) - g(\mathbf{L}^*) \leq 8C_g k^{-1}$. We specialize Hazan's algorithm it to our framework as follows (below, note that $\mathbf{A} = \mathbf{K}_\eta \mathbf{C}$, $\mathbf{B} = \mathbf{C}^T \mathbf{A}$):

- Using **bounded trace constraints**, $\text{trace}(\mathbf{L}) \leq \tau$, instead of unit trace is more meaningful for our setting. The following modified updates optimize over $\mathcal{S}_+^n(\tau)$: $\mathbf{L}_{k+1} = \mathbf{L}_k + \alpha_k (\tau \mathbf{v}_k \mathbf{v}_k^T - \mathbf{L}_k)$, where \mathbf{v}_k is reset to the zero matrix if the smallest eigenvalue is positive.
- The **gradient for our objective** is: $\nabla g(\mathbf{L}) = \mathbf{G} + \mathbf{G}^T - \text{diag}(\mathbf{G})$ where $\mathbf{G} = \nabla \Gamma(\mathbf{L}) + \lambda \mathbf{B} + 2\mathbf{A}^T \mathbf{A} \mathbf{L} - 2\mathbf{A}^T \mathbf{Y}$ and $\text{diag}(\cdot)$ assembles the diagonal entries of its argument into a diagonal matrix.
- For $\Gamma(\mathbf{L}) = \rho \|\mathbf{L}\|_{fro}^2$, instead of using Hazan's line search parameter α_k , we do **exact line search** along the direction $\mathbf{P} = \tau \mathbf{v}_k \mathbf{v}_k^T - \mathbf{L}_k$ which leads to a closed form expression: $\alpha_k = \frac{\text{trace}((\frac{1}{T} \mathbf{A} \mathbf{L} - \mathbf{Y})^T \mathbf{A} \mathbf{P} + \rho \mathbf{L}^T \mathbf{P} + \frac{1}{T} \lambda \mathbf{B} \mathbf{P})}{\text{trace}(\rho \mathbf{P}^T \mathbf{P} + \frac{1}{T} \mathbf{P}^T \mathbf{A}^T \mathbf{A} \mathbf{P})}$. If this value exceeds 1.0, we set $\alpha_k = 1$ since the objective function restricted to a line is a simple quadratic.

- The Hessian of g , with $\Gamma(\mathbf{L}) = \rho \|\mathbf{L}\|_{fro}^2$, in vectorized notation is $\mathbf{I}_{n^2} \otimes \mathbf{A}^T \mathbf{A} + \rho \mathbf{I}_{n^4}$, whose maximum eigenvalue, assuming $\sum_j \eta_j \leq 1$ as for l_1 norms, can be upper bounded by ρ plus $\sigma_1^2 = \max_j \sigma_{1j}$ where σ_{1j} refers to the maximum eigenvalue of \mathbf{K}_j . Following [12], we get $C_g \leq 2\tau(\sigma_1^2 + \rho)$ which implies the bound: $g(\mathbf{L}_{k+1}) - g(\mathbf{L}^*) \leq 16\tau(\sigma_1^2 + \rho)k^{-1}$.

3.1 Rademacher Complexity Results

The notion of Rademacher complexity is readily generalizable to vector-valued hypothesis spaces [18]. Let \mathcal{H} be a class of functions $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} \subset \mathbb{R}^n$. Let $\boldsymbol{\sigma} \in \mathbb{R}^n$ be a vector of independent Rademacher variables, and similarly define the matrix $\Sigma = [\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_l] \in \mathbb{R}^{n \times l}$. The empirical Rademacher complexity of the vector-valued class \mathcal{H} is the function $\hat{R}_l(\mathcal{H})$ defined as $\hat{R}_l(\mathcal{H}) = \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}} \sum_{i=1}^l \boldsymbol{\sigma}_i^T f(\mathbf{x}_i) \right]$. For general matrix-valued multiple kernel learning algorithms, we consider hypothesis spaces of the form $\mathcal{H} = \{f = \sum_{j=1}^m f_j, f_j \in \mathcal{H}_{\vec{k}_j}\}$, and with the l_p norm $\|f\|_{l(\mathcal{H})} = \inf_{f=\sum_j f_j} \left\| \|f_1\|_{\mathcal{H}_{\vec{k}_1}}, \dots, \|f_m\|_{\mathcal{H}_{\vec{k}_m}} \right\|_p$.

Theorem 3.1. Consider the hypothesis class $\mathcal{H}_\lambda^p = \{f \in \mathcal{H} : \|f\|_{l_p(\mathcal{H})} \leq \lambda\}$.

- For $p > 1$, the Rademacher complexity of hypothesis class \mathcal{H}_λ^p can be upper-bounded as follows:
 $\hat{R}_l(\mathcal{H}_\lambda^p) \leq \frac{\lambda \|\mathbf{u}\|_q}{l}$ where $\mathbf{u} = \left[\sqrt{\text{trace}(\vec{\mathbf{K}}_1)}, \dots, \sqrt{\text{trace}(\vec{\mathbf{K}}_m)} \right]$, and q is such that $\frac{1}{p} + \frac{1}{q} = 1$.
- For $p > 1$ and the special case of separable kernels $\vec{k}_i(\mathbf{x}, \mathbf{z}) = k_i(\mathbf{x}, \mathbf{z})\mathbf{L}$ such that $\sup_{\mathbf{x}} k_i(\mathbf{x}, \mathbf{x}) \leq \kappa$ and $\text{trace}(\mathbf{L}) \leq \tau$ we have $\hat{R}_l(\mathcal{H}_\lambda^p) \leq \lambda m^{1/q} \sqrt{\frac{\kappa \tau}{l}}$.
- For $p = 1$ we obtain, $\hat{R}_l(\mathcal{H}_\lambda^1) \leq \frac{\lambda \sqrt{\eta_0 q}}{l} \|\mathbf{u}\|_q$ for any $q > 0$ where $\eta_0 = \frac{23}{22}$.
- For $p = 1$ and the special case of separable kernels $\vec{k}_i(\mathbf{x}, \mathbf{z}) = k_i(\mathbf{x}, \mathbf{z})\mathbf{L}$ such that $\sup_{\mathbf{x}} k_i(\mathbf{x}, \mathbf{x}) \leq \kappa$ and $\text{trace}(\mathbf{L}) \leq \tau$ we have $\hat{R}_l(\mathcal{H}_\lambda^1) \leq \frac{\lambda \sqrt{\eta_0 2e \log(m) \kappa \tau}}{\sqrt{l}}$.

The proofs are provided in Appendix B. The above results straightforwardly lead to generalization bounds. They extend the results of [10].

4 High Dimensional Non-linear Causal Inference

Here, our goal is to show how high-dimensional causal inference tasks can be naturally cast as sparse function estimation problems within our framework, leading to novel nonlinear extensions of Grouped Graphical Granger Causality techniques (see [25, 16] and references therein). In this setting, there is an interconnected system of N distinct sources of high dimensional time series data which we denote as $\mathbf{x}_t^i \in \mathbb{R}^{d_i}, i = 1 \dots N$. The system is observed from time $t = 1$ to $t = T$, and the goal is to infer the causal relationships between the sources. Let G denote the adjacency matrix of the unknown causal interaction graph where $G_{ij} = 1$ implies that source i causally influences source j . In 1980, Clive Granger gave an operational definition for Causality:

Granger Causality: A subset of sources $A_i = \{j : G_{ij} = 1\}$ is said to causally influence source i , if the past values of the time series collective associated with the source set A_i is predictive of the future evolution of the time series associated with source i , with statistical significance, and more so than the past values of i alone.

A practical appeal of this definition is that it links causal inference to prediction, with the caveat that causality insights are bounded by the quality of the underlying predictive model. Furthermore, the prior knowledge that the underlying causal interactions are highly selective makes sparsity a meaningful prior to use. Prior work in this direction has focused on linear models [25, 16] while many, if not most, natural systems often involve nonlinear interactions. To apply our framework to such problems, we model the system as: $\mathbf{x}_t^i = f^i(\mathbf{x}_t^1, \mathbf{x}_{t-1}^1 \dots \mathbf{x}_{t-l}^1, \dots, \mathbf{x}_t^N, \mathbf{x}_{t-1}^N \dots \mathbf{x}_{t-l}^N)$, where l is a lag parameter and $f^i \in \mathcal{H}(\mathcal{D}_{\mathbf{L}^i})$ where \mathbf{L}^i is the output kernel matrix, and we work with a dictionary of subsystem-specific input kernels: $\mathcal{D} = \cup_{r=1}^N \{k_{rs}\mathbf{L}\}_{s=1}^{m_j}$ where k_{rs} is one of a choice of m_r scalar kernel functions that only depend on the past values of source r , i.e., $\mathbf{x}_1^r \dots \mathbf{x}_{t-l}^r$. Then, by imposing functional sparsity in the estimation of f^i by solving Eqn. 5 using $\|f\|_{l_1(\mathcal{H}_{\mathbf{D}_{\mathbf{L}}})}$

regularizer, we get a novel nonparameteric implementation for Granger Causality. In particular, if η_{rs}^i corresponds to the weight on kernel k_{rs} for the estimated function f^i , then we set $G_{ij} = \sum_{s=1}^{m_j} \eta_{js}^i$. In addition to recovering the causal graph in this way, the estimated output kernel matrix \mathbf{L}^i captures the within-source temporal dependencies.

5 Empirical Studies

VAR Modeling in Financial Time Series Analysis: We start with a small dataset of weekly log returns of 9 stocks from 2004, studied in [28, 23] in the context of linear multivariate regression with output covariance estimation techniques. We consider first-order vector autoregressive (VAR) models of the form $\mathbf{x}_t = f(\mathbf{x}_{t-1})$ where \mathbf{x}_t corresponds to the 9-dimensional vector of log-returns for the 9 companies at week t and the function f is estimated by solving Eqn. 5. Our experimental protocol is exactly the same as [28, 23]: data is split evenly into a training and a test set and the regularization parameter λ is chosen by 10-fold cross-validation. All other parameters are left at their default values. We generated a dictionary of 117 Gaussian kernels defined by univariate Gaussian kernels on each of the 9 dimensions with 13 varying bandwidths. Results are shown in Figure 1 (table on the left) where we compare our methods in terms of mean test RMSE against standard linear regression (OLS) and linear Lasso independently applied to each output coordinate, and the sparse multivariate regression with covariance estimation approaches of [23, 28], labeled MRCE and FES respectively. We see that joint input and output kernel learning (labeled IOKL) yields the best return prediction model reported to date on this dataset. As expected, it outperforms models obtained by leaving output kernel matrix fixed as the identity and only optimizing scalar kernels (IKL), or only optimizing the output kernel for fixed choices of scalar kernel (OKL). Of the 117 kernels, 13 have 97% of the mass in the learnt scalar kernel combination. In Figure 1 we also show the learnt output kernel \mathbf{L} , which notably captures strong similarity between large automobile companies: Ford and GM.

	OLS	Lasso	MRCE	FES	IKL	OKL	IOKL
Walmart	0.98	0.42	0.41	0.40	0.43	0.43	0.44
Exxon	0.39	0.31	0.31	0.29	0.32	0.31	0.29
GM	1.68	0.71	0.71	0.62	0.62	0.59	0.47
Ford	2.15	0.77	0.77	0.69	0.56	0.48	0.36
GE	0.58	0.45	0.45	0.41	0.41	0.40	0.37
ConocoPhillips	0.98	0.79	0.79	0.79	0.81	0.80	0.76
Citigroup	0.65	0.66	0.62	0.59	0.66	0.62	0.58
IBM	0.62	0.49	0.49	0.51	0.47	0.50	0.42
AIG	1.93	1.88	1.88	1.74	1.94	1.87	1.79
Average	1.11	0.72	0.71	0.67	0.69	0.67	0.61

Walmart	0.26	0.11	0.60	0.76	0.26	0.17	0.25	0.22	0.27
Exxon	0.11	0.27	0.19	0.24	0.23	0.31	0.16	0.17	0.31
GM	0.60	0.19	2.22	2.67	0.82	0.35	0.79	0.68	0.76
Ford	0.76	0.24	2.67	3.72	0.99	0.52	0.75	0.63	0.96
GE	0.26	0.23	0.82	0.99	0.46	0.36	0.38	0.35	0.48
ConocoPhillips	0.17	0.31	0.35	0.52	0.36	0.55	0.18	0.21	0.46
Citigroup	0.25	0.16	0.79	0.75	0.38	0.18	0.48	0.42	0.37
IBM	0.22	0.17	0.68	0.63	0.35	0.21	0.42	0.46	0.36
AIG	0.27	0.31	0.76	0.96	0.48	0.46	0.37	0.36	0.59
	Walmart	Exxon	GM	Ford	GE	ConocoPhillips	Citigroup	IBM	AIG

Figure 1: RMSE (top) and Output kernel \mathbf{L} (bottom)

Scalability and Numerical Behaviour: Our main interest here is to observe the classic tradeoff in numerical optimization between running few, but expensive steps versus executing several cheap

iterations. We use a 102-class image categorization dataset – Caltech-101 – which has been very well studied in the multiple kernel learning literature [11, 27, 13]. There are 30 training images per category for a total of 3060 training images, and 1355 test images. Targets are 102-dimensional class indicator vectors. We define a dictionary of kernels using 10 scalar-valued kernels precomputed from visual features and made publically available by the authors of [27], for 3 training/test splits. From previous studies, it is well known that all underlying visual features contribute to object discrimination on this dataset and hence non-sparse multiple kernel learning with $l_p, p > 1$ norms are more effective. We therefore set $p = 1.7$ and $\lambda = 0.001$ without any further tuning, since their choice is not central to our main goals in this experiment. We vary the stopping criteria for our CG-based Sylvester solver (cg_ϵ) and the number of iterations (sdp_{iter}) allowed in the Sparse SDP solver, for the **C** and **L** subproblems respectively. Note that the closed form η updates for l_p norms take negligible time. We compare our algorithms with an implementation in which each subproblem is solved exactly using an eigendecomposition based Sylvester solver for **C**, and unconstrained updates for **L** developed in [11], respectively. To make comparisons meaningful, we set τ to a large value so that the optimization over $\mathbf{L} \in \mathcal{S}_+^n(\tau)$ effectively corresponds to unconstrained minimization over the entire psd cone \mathcal{S}_+^n . In Figure 2, we report the improvement in objective function and classification accuracy as a function of time (upto 1 hour). We see that insufficient progress is made in both extremes: when either the degree of inexactness is intolerable ($cg_\epsilon = 0.1, sdp_{iter} = 100$) or when subproblems are solved to very high precision ($cg_\epsilon = 1e-6, sdp_{iter} = 3000$). Our solvers are far more efficient than eigendecomposition based implementation that takes an exorbitant amount of time per iteration for exact solutions. Approximate solvers at appropriate precision (e.g., $cg_\epsilon = 0.01, sdp_{iter} = 1000$) make very rapid progress and return high accuracy models in just a few minutes. In fact, averaged over the three training/test splits, the classification accuracy obtained is $79.43\% \pm 0.67$ which is highly competitive with state of the art results reported on this dataset, with the kernels used above. For example, [27] report $78.2\% \pm 0.4$, [13] report $77.7\% \pm 0.3$ and [11] report 75.36% .

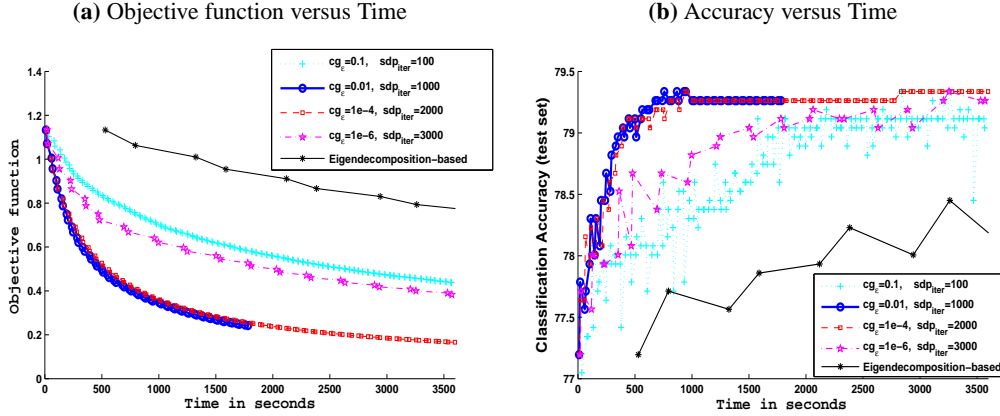
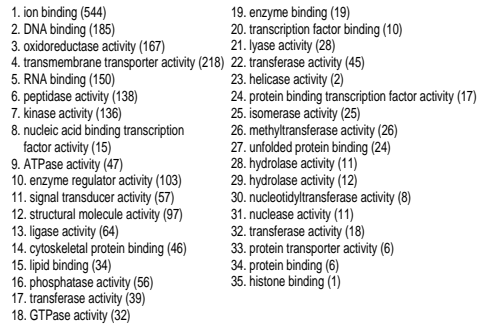


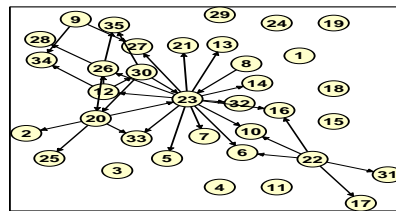
Figure 2: Caltech 101 Image Categorization

Causal Inference of Gene Networks: We use time-course gene expression microarray data measured during the full life cycle of *Drosophila melanogaster* [2]. The expression levels of 4028 genes are simultaneously measured at 66 time points corresponding to various developmental stages. We extracted time series data for 2397 unique genes, and grouped them into 35 functional groups based on their gene ontologies listed in Figure 3(b) together with the number of genes in parenthesis. The goal is to infer causal interactions between functional groups, as well obtain insight on within-group relationships between genes. This is an instance of the setting described in Section 4. We conducted 4 sets of experiments: with linear and nonlinear dictionaries (Gaussian kernels with 13 choices of bandwidths per group), and with or without output kernel learning. We use the parameters $\lambda = 0.001$ and time lag of 7. Hold-out RMSE from the four models is shown in Figure 3 (a) for the 35 groups. Clearly, nonlinear models with both input and output kernel learning give the best predictive performance implying greater reliability in the implied causal graphs. In consultation with a professional biologist, we analyzed the causal graphs uncovered by our approach. The difference between the graphs uncovered by linear and nonlinear models (shown in Figure 3 (c) and (d)) is intriguing. In particular our nonlinear model uncovered the centrality of a key cellular enzymatic

(b) Function Gene Groups



(d) Causal Graph (Nonlinear)



activity, that of helicase, which was not recognized by the linear model. In contrast, the central nodes in the linear model are related to membranes (lipid binding and gtpase activity). Nucleic acid binding transcription factor activity and transcription factor binding are both related to the helicase activity, which is consistent with biological knowledge of them being tightly coupled. This was not captured in the linear model. Molecular chaperone functions, which connect ATPase activity and unfolded protein binding, was successfully identified by our model, while the linear model failed to recognize its relevance. It is less likely that unfolded protein and lipid activity should be linked as suggested by the linear model. In addition, via output kernel estimation, our model also provides insight on the conditional dependencies within genes for each functional group, e.g., Fig 3(e) for the *unfolded protein binding* group.

References

- 9

- [4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity inducing penalties. *Foundations and Trends in Machine Learning*, 2011.
- [5] A. Berline and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer, 2004.
- [6] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Providence, RI, 2003.
- [7] Peter Buhlmann and Sara Van De Geer. *Statistics for High Dimensional Data*. Springer, 2010.
- [8] A. Caponnetto, M. Pontil, C. Micchelli, and Y. Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 9:1615–1646, 2008.
- [9] K. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms*, 2010.
- [10] C. Cortes, M. Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *ICML*, 2010.
- [11] Peter Gehler Gianluigi Pillonetto Francesco Dinuzzo, Cheng Soon Ong. Learning output kernels with block coordinate descent. In *ICML*, 2011.
- [12] B. Gartner and J. Matousek. *Approximation Algorithms and Semi-definite Programming*. Springer-Verlag, Berlin Heidelberg, 2012.
- [13] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.
- [14] C. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2:329–352, 1980.
- [15] E. Hazan. Sparse approximate solutions to semi-definite programs. In *LATIN*, 2008.
- [16] Aurelie C. Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. Grouped graphical granger modeling methods for temporal causal modeling. In *KDD*, pages 577–586, 2009.
- [17] H. Lutkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2007.
- [18] A. Maurer. The rademacher complexity of linear transformation classes. In *Proceedings of the Conference on Learning Theory (COLT)*, 2006.
- [19] C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.
- [20] C. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.
- [21] J. Peters, D. Janzing, A. Gretton, and B. Schoelkopf. Detecting the direction of causal time series. In *Proceedings of the International Conference on Machine Learning*, 2009.
- [22] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.
- [23] A. J. Rothman, E. Levina, and J. Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 1:947–962, 2010.
- [24] L. Schwartz. Sous-espaces hilbertiens d’espaces vectoriels topologiques et noyaux associés (noyaux reproduisants). *J. Analyse Math.*, 13:115–256, 1964.
- [25] Ali Shojaie and George Michailidis. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, September 2010.
- [26] R. Tomioka and T. Suzuki. Regularization strategies and empirical bayesian learning for mkl. In *NIPS Workshops 2010*, 2010.
- [27] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *International Conference on Computer Vision*, 2009.
- [28] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 2010.

Appendix A.

Proposition 1 (Vector-valued extension of sum of reproducing kernels (section 6) Theorem in [3]). *Let $\vec{k}_1 \dots \vec{k}_p$ be operator-valued reproducing kernels of a dictionary of RKHSs $\mathcal{D} = \{\mathcal{H}_{\vec{k}_1} \dots \mathcal{H}_{\vec{k}_p}\}$ mapping $\mathcal{X} \mapsto \mathcal{Y}$ with respective norms $\|\cdot\|_{\mathcal{H}_{\vec{k}_1}} \dots \|\cdot\|_{\mathcal{H}_{\vec{k}_p}}$. Then $\vec{k}_{\mathcal{D}} = \sum_{i=1}^p \lambda_i \vec{k}_i$, with $\lambda_i > 0$, $i = 1 \dots p$, is the reproducing kernel of the space $\mathcal{H}_{\mathcal{D}} = \mathcal{H}_{\vec{k}_1} \oplus \dots \oplus \mathcal{H}_{\vec{k}_p}$ with the norm $\|\cdot\|_{\mathcal{H}_{\mathcal{D}}}$ given by*

$$\|f\|_{\mathcal{H}_{\mathcal{D}}}^2 = \arg \min_{f = \sum_{i=1}^p f_i, f_i \in \mathcal{H}_{\vec{k}_i}} \sum_{i=1}^p \frac{\|f_i\|^2}{\lambda_i} \quad (8)$$

Proof. First, note that if the Theorem holds for $p = 2$, it can be inductively applied to furnish a proof for a general p . So without loss of generality, we assume $p = 2$. As in Section 6 of [3] or Theorem 5 in [5], we start by introducing the product space,

$$\mathcal{F} = \mathcal{H}_{\vec{k}_1} \times \mathcal{H}_{\vec{k}_2}$$

and an inner product on \mathcal{F} defined by,

$$\langle (f^1, f^2), (g^1, g^2) \rangle_{\mathcal{F}} = \frac{1}{\lambda_1} \langle f^1, g^1 \rangle_{\mathcal{H}_{\vec{k}_1}} + \frac{1}{\lambda_2} \langle f^2, g^2 \rangle_{\mathcal{H}_{\vec{k}_2}}$$

Define the map $u : \mathcal{F} \mapsto \mathcal{H}_{\mathcal{D}}$ by $u(f^1, f^2) = f^1 + f^2$. Due to completeness of $\mathcal{H}_{\vec{k}_1}, \mathcal{H}_{\vec{k}_2}$, the map's kernel $N := u^{-1}(0)$ is a closed subspace of \mathcal{F} and thus its orthogonal complement is also a closed subspace. We can consider N^\perp as a Hilbert space with the inner product that is the natural restriction of $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ to N . Define $v : N^\perp \mapsto \mathcal{H}_{\mathcal{D}}$ as the restriction of u to N^\perp . Then v is a bijection, and we define an inner product on $\mathcal{H}_{\mathcal{D}}$ by

$$\langle f, g \rangle_{\mathcal{H}_{\mathcal{D}}} = \langle v^{-1}(f), v^{-1}(g) \rangle_{\mathcal{F}}$$

In other words, $\mathcal{H}_{\mathcal{D}}$ is a Hilbert space isomorphic to N^\perp . Fix any $f \in \mathcal{H}_{\mathcal{D}}$ and note that $u^{-1}(f) = \{v^{-1}(f) + n | n \in N\}$. Since $v^{-1}(f)$ and N are orthogonal, it is clear by the Pythagorean theorem that $v^{-1}(f)$ is the element of $u^{-1}(f)$ with minimum norm. Thus,

$$\|f\|_{\mathcal{H}_{\mathcal{D}}}^2 = \|v^{-1}(f)\|_{\mathcal{F}}^2 = \min_{(f^1, f^2) \in u^{-1}f} \|(f^1, f^2)\|_{\mathcal{F}}^2 = \frac{\|f^1\|_{\mathcal{H}_1}^2}{\lambda_1} + \frac{\|f^1\|_{\mathcal{H}_2}^2}{\lambda_2}$$

We see that the inner product on $\mathcal{H}_{\mathcal{D}}$ induces the norm claimed in the theorem statement.

For any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, it is clear that $\vec{k}_{\mathcal{D}}$ is a symmetric positive semi-definite function since its a conic combination of base kernels, and that $\vec{k}_{\mathcal{D}}(x, \cdot)y = \lambda_1 \vec{k}_1(x, \cdot)y + \lambda_2 \vec{k}_2(x, \cdot)y \in \mathcal{H}_1 \oplus \mathcal{H}_2 = \mathcal{H}_{\mathcal{D}}$. So we just need to verify the reproducing property: $\langle f, \vec{k}_{\mathcal{D}}(x, \cdot)y \rangle_{\mathcal{H}_{\mathcal{D}}} = \langle f(x), y \rangle_{\mathcal{Y}}$. Towards this, take any $f \in \mathcal{H}_{\mathcal{D}}$ and let $(f^1, f^2) = v^{-1}(f)$. Also, let $(\vec{h}^1, \vec{h}^2) = v^{-1}(\vec{k}_{\mathcal{D}}(x, \cdot)y)$. Next observe that,

$$(\vec{h}^1 - \lambda_1 \vec{k}_1(x, \cdot)y + \vec{h}^2 - \lambda_2 \vec{k}_2(x, \cdot)y) = \vec{k}_{\mathcal{D}}(x, \cdot)y - \vec{k}_{\mathcal{D}}(x, \cdot)y = 0$$

Hence, $(\vec{h}^1 - \lambda_1 \vec{k}_1(x, \cdot)y, \vec{h}^2 - \lambda_2 \vec{k}_2(x, \cdot)y) \in N$ and therefore its inner product in \mathcal{F} with (f^1, f^2) equals 0. We have,

$$\begin{aligned} \langle (f^1, f^2), (\vec{h}^1 - \lambda_1 \vec{k}_1(x, \cdot)y, \vec{h}^2 - \lambda_2 \vec{k}_2(x, \cdot)y) \rangle_{\mathcal{F}} &= 0 \\ \frac{1}{\lambda_1} \langle f^1, (\vec{h}^1 - \lambda_1 \vec{k}_1(x, \cdot)y) \rangle_{\mathcal{H}_{\vec{k}_1}} + \frac{1}{\lambda_2} \langle f^2, (\vec{h}^2 - \lambda_2 \vec{k}_2(x, \cdot)y) \rangle_{\mathcal{H}_{\vec{k}_2}} &= 0 \\ \frac{1}{\lambda_1} \langle f^1, \vec{h}^1 \rangle_{\mathcal{H}_{\vec{k}_1}} + \frac{1}{\lambda_2} \langle f^2, \vec{h}^2 \rangle_{\mathcal{H}_{\vec{k}_2}} &= \langle f^1, \vec{k}_1(x, \cdot)y \rangle_{\mathcal{H}_{\vec{k}_1}} + \langle f^2, \vec{k}_2(x, \cdot)y \rangle_{\mathcal{H}_{\vec{k}_2}} \end{aligned} \quad (9)$$

We will use the last equality in the final steps below together with reproducing properties of \vec{k}_1 and \vec{k}_2 .

$$\begin{aligned} \langle f \vec{k}_{\mathcal{D}}(x, \cdot)y \rangle_{\mathcal{H}_{\mathcal{D}}} &= \langle v^{-1}(f), v^{-1}(\vec{k}_{\mathcal{D}}(x, \cdot)y) \rangle_{\mathcal{F}} \\ &= \langle (f^1, f^2), (\vec{h}^1, \vec{h}^2) \rangle_{\mathcal{F}} \\ &= \frac{1}{\lambda_1} \langle f^1, \vec{h}^1 \rangle_{\mathcal{H}_{\vec{k}_1}} + \frac{1}{\lambda_2} \langle f^2, \vec{h}^2 \rangle_{\mathcal{H}_{\vec{k}_2}} \\ &= \langle f^1, \vec{k}_1(x, \cdot)y \rangle_{\mathcal{H}_{\vec{k}_1}} + \langle f^2, \vec{k}_2(x, \cdot)y \rangle_{\mathcal{H}_{\vec{k}_2}} \\ &= \langle f^1(x), y \rangle_{\mathcal{Y}} + \langle f^2(x), y \rangle_{\mathcal{Y}} \\ &= \langle f(x), y \rangle_{\mathcal{Y}} \end{aligned}$$

Appendix B: Rademacher Complexity

The notion of Rademacher complexity is readily generalizable to vector-valued hypothesis spaces [18]. Let \mathcal{H} be a class of functions $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} \subset \mathbb{R}^n$. Let $\sigma \in \mathbb{R}^n$ be a vector of independent Rademacher

variables, and similarly define the matrix $\Sigma = [\sigma_1, \dots, \sigma_l] \in \mathbb{R}^{n \times l}$. The empirical Rademacher complexity of the class \mathcal{H} is the function $\hat{R}_l(\mathcal{H})$ defined as

$$\hat{R}_l(\mathcal{H}) = \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}} \sum_{i=1}^l \sigma_i^T f(\mathbf{x}_i) \right]$$

We first focus on vector-valued RKHS and obtain the following result.

Theorem 5.1. *Let \mathcal{H} be a vector-valued RKHS with associated kernel \vec{k} . Consider the hypothesis class $\mathcal{H}_\lambda = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}}^2 \leq \lambda\}$. The Rademacher complexity of hypothesis class \mathcal{H}_λ can be upperbounded as follows.*

$$\hat{R}_l(\mathcal{H}_\lambda) \leq \frac{\sqrt{\lambda \text{trace}(\vec{\mathbf{K}})}}{l},$$

where K is the Gram matrix of the kernel \vec{k} . For the special case of separable kernels $\vec{k}(\mathbf{x}, \mathbf{z}) = k(\mathbf{x}, \mathbf{z})\mathbf{L}$ such that $\sup_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}) \leq \kappa$ and $\text{trace}(\mathbf{L}) \leq \tau$ we have

$$\hat{R}_l(\mathcal{H}_\lambda) \leq \sqrt{\frac{\lambda \kappa \tau}{l}}.$$

Proof. Recall that by the Reproducing property we have $\sigma^T f(\mathbf{x}) = \langle f, \vec{k}(\mathbf{x}, \cdot) \sigma \rangle_{\mathcal{H}}$. Then we get

$$\begin{aligned} \hat{R}_l(\mathcal{H}_\lambda) &= \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}_\lambda} \langle f, \sum_{i=1}^l \vec{k}(\mathbf{x}_i, \cdot) \sigma_i \rangle_{\mathcal{H}} \right] \\ &\leq \frac{1}{l} \sup_{f \in \mathcal{H}_\lambda} \|f\|_{\mathcal{H}} \mathbb{E}_\Sigma \left\| \sum_{i=1}^l \vec{k}(\mathbf{x}_i, \cdot) \sigma_i \right\|_{\mathcal{H}} \\ &\leq \frac{\sqrt{\lambda}}{l} \mathbb{E}_\Sigma \left\| \sum_{i=1}^l \vec{k}(\mathbf{x}_i, \cdot) \sigma_i \right\|_{\mathcal{H}} \\ &= \frac{\sqrt{\lambda}}{l} \mathbb{E}_\Sigma \sqrt{\left\langle \sum_{i=1}^l \vec{k}(\mathbf{x}_i, \cdot) \sigma_i, \sum_{j=1}^l \vec{k}(\mathbf{x}_j, \cdot) \sigma_j \right\rangle_{\mathcal{H}}} \\ &= \frac{\sqrt{\lambda}}{l} \mathbb{E}_\Sigma \sqrt{\sum_{i,j} \sigma_i^T \vec{k}(\mathbf{x}_i, \mathbf{x}_j) \sigma_j} \\ &\leq \frac{\sqrt{\lambda}}{l} \sqrt{\mathbb{E}_\Sigma \text{trace}(\Sigma^T \vec{\mathbf{K}} \Sigma)} \\ &\leq \frac{\sqrt{\lambda}}{l} \sqrt{\text{trace}(\mathbb{E}_\Sigma(\Sigma \Sigma^T) \vec{\mathbf{K}})} \\ &\leq \frac{\sqrt{\lambda \text{trace}(\vec{\mathbf{K}})}}{l} \end{aligned}$$

Now for the special case of separable kernels we obtain

$$\hat{R}_l(\mathcal{H}_\lambda) \leq \sqrt{\frac{\lambda \kappa \tau}{l}}.$$

□

We now consider the multiple kernel learning case.

Theorem 5.2. *Let $\mathcal{H} = \{f = \sum_{j=1}^m f_j, f_j \in \mathcal{H}_{\vec{k}_j}\}$, and define the norm*

$$\|f\|_l(\mathcal{H}) = \inf_{f = \sum_j f_j} \|\|f_1\|_{\mathcal{H}_1}, \dots, \|f_m\|_{\mathcal{H}_m}\|_p.$$

Consider the hypothesis class $\mathcal{H}_\lambda^p = \{f \in \mathcal{H} : \|f\|_{l_p(\mathcal{H})} \leq \lambda\}$. For $p > 1$, the Rademacher complexity of hypothesis class \mathcal{H}_λ^p can be upperbounded as follows.

$$\hat{R}_l(\mathcal{H}_\lambda^p) \leq \frac{\lambda \|\mathbf{u}\|_q}{l}$$

where $\mathbf{u} = \left[\sqrt{\text{trace}(\vec{\mathbf{K}}_1)}, \dots, \sqrt{\text{trace}(\vec{\mathbf{K}}_m)} \right]$, and q is such that $\frac{1}{p} + \frac{1}{q} = 1$.

For the special case of separable kernels $\vec{k}_i(\mathbf{x}, \mathbf{z}) = k_i(\mathbf{x}, \mathbf{z})\mathbf{L}$ such that $\sup_{\mathbf{x}} k_i(\mathbf{x}, \mathbf{x}) \leq \kappa$ and $\text{trace}(\mathbf{L}) \leq \tau$ we have

$$\hat{R}_l(\mathcal{H}_\lambda^p) \leq \lambda m^{1/q} \sqrt{\frac{\kappa\tau}{l}}.$$

For $p = 1$ we obtain,

$$\hat{R}_l(\mathcal{H}_\lambda^1) \leq \frac{\lambda\sqrt{\eta_0 q}}{l} \|\mathbf{u}\|_q$$

for any $q > 0$.

For the special case of separable kernels $\vec{k}_i(\mathbf{x}, \mathbf{z}) = k_i(\mathbf{x}, \mathbf{z})\mathbf{L}$ such that $\sup_{\mathbf{x}} k_i(\mathbf{x}, \mathbf{x}) \leq \kappa$ and $\text{trace}(\mathbf{L}) \leq \tau$ we have

$$\hat{R}_l(\mathcal{H}_\lambda^1) \leq \frac{\lambda\sqrt{\eta_0 2e \log(m)\kappa\tau}}{\sqrt{l}}$$

Proof. We first focus on $p > 1$. We again make use of the reproducible property of kernels.

$$\begin{aligned} \hat{R}_l(\mathcal{H}_\lambda^p) &= \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}_\lambda^p} \sum_{i=1}^l \boldsymbol{\sigma}_i^T \sum_{j=1}^m f_j(\mathbf{x}_i) \right] \\ &= \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}_\lambda^p} \sum_{j=1}^m \langle f_j, \sum_{i=1}^l \vec{k}_j(\mathbf{x}, \cdot) \boldsymbol{\sigma}_i \rangle_{\mathcal{H}} \right] \\ &\leq \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}_\lambda^p} \sum_{j=1}^m \|f_j\|_{\mathcal{H}_j} \left\| \sum_{i=1}^l \vec{k}_j(\mathbf{x}, \cdot) \boldsymbol{\sigma}_i \right\|_{\mathcal{H}_j} \right] \\ &= \frac{1}{l} \sup_{f \in \mathcal{H}_\lambda^p} \sum_{j=1}^m \|f_j\|_{\mathcal{H}_j} \mathbb{E}_\Sigma \left[\left\| \sum_{i=1}^l \vec{k}_j(\mathbf{x}, \cdot) \boldsymbol{\sigma}_i \right\|_{\mathcal{H}_j} \right] \\ &= \frac{1}{l} \sup_{f \in \mathcal{H}_\lambda^p} \sum_j \|f_j\|_{\mathcal{H}_j} \sqrt{\text{trace}(\vec{\mathbf{K}}_j)} \\ &\leq \frac{1}{l} \sup_{f_1, \dots, f_m: \|\|f_1\|_{\mathcal{H}_1}, \dots, \|f_m\|_{\mathcal{H}_m}\|_p \leq \lambda} \sum_j \|f_j\|_{\mathcal{H}_j} \sqrt{\text{trace}(\vec{\mathbf{K}}_j)} \end{aligned}$$

Let $\mathbf{u} = \left[\sqrt{\text{trace}(\vec{\mathbf{K}}_1)}, \dots, \sqrt{\text{trace}(\vec{\mathbf{K}}_m)} \right]$ and $\mathbf{v} = [\|f_1\|_{\mathcal{H}_1}, \dots, \|f_m\|_{\mathcal{H}_m}]$. With this notation we have

$$\begin{aligned} \hat{R}_l(\mathcal{H}_\lambda^p) &\leq \frac{1}{l} \sup_{\|\mathbf{v}\|_p \leq \lambda} \mathbf{v}^T \mathbf{u} \\ &\leq \frac{1}{l} \sup_{\|\mathbf{v}\|_p \leq \lambda} \|\mathbf{v}\|_p \|\mathbf{u}\|_q \\ &\leq \frac{\lambda \|\mathbf{u}\|_q}{l}, \end{aligned}$$

where the first inequality is a direct consequence of Hölder's inequality.

Now for the special case of separable kernels we have $\|\mathbf{u}\|_q \leq \left(\sum_{i=1}^m \sqrt{l\kappa\tau}^q \right)^{1/q}$ and thus conclude

$$\hat{R}_l(\mathcal{H}_\lambda^p) \leq \lambda m^{1/q} \sqrt{\frac{\kappa\tau}{l}}$$

We now focus on the special case where $p = 1$

$$\begin{aligned}
\hat{R}_l(\mathcal{H}_\lambda^1) &= \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}_\lambda^1} \sum_{i=1}^l \sigma_i^T \sum_{j=1}^m f_j(\mathbf{x}_i) \right] \\
&= \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}_\lambda^1} \sum_{j=1}^m \langle f_j, \sum_{i=1}^l \vec{k}_j(\mathbf{x}, \cdot) \sigma_i \rangle_{\mathcal{H}} \right] \\
&\leq \frac{1}{l} \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{H}_\lambda^1} \sum_{j=1}^m \|f_j\|_{\mathcal{H}_j} \left\| \sum_{i=1}^l \vec{k}_j(\mathbf{x}, \cdot) \sigma_i \right\|_{\mathcal{H}_j} \right]
\end{aligned}$$

Let

$$\mathbf{w} = \left[\left\| \sum_{i=1}^l \vec{k}_1(\mathbf{x}, \cdot) \sigma_i \right\|_{\mathcal{H}_1}, \dots, \left\| \sum_{i=1}^l \vec{k}_m(\mathbf{x}, \cdot) \sigma_i \right\|_{\mathcal{H}_m} \right]$$

and $\mathbf{v} = [\|f_1\|_{\mathcal{H}_1}, \dots, \|f_m\|_{\mathcal{H}_m}]$. With this notation we have

$$\begin{aligned}
\hat{R}_l(\mathcal{H}_\lambda^1) &\leq \frac{1}{l} \mathbb{E}_\Sigma \sup_{\|\mathbf{v}\|_1 \leq \lambda} \mathbf{v}^T \mathbf{w} \\
&\leq \frac{1}{l} \mathbb{E}_\Sigma \sup_{\|\mathbf{v}\|_1 \leq \lambda} \|\mathbf{v}\|_p \|\mathbf{w}\|_q \quad \text{for any } p, q : 1/p + 1/q = 1 \\
&\leq \frac{\lambda}{l} \mathbb{E}_\Sigma \|\mathbf{w}\|_q \quad \text{using } \|\mathbf{v}\|_p \leq \|\mathbf{v}\|_1 \\
&= \frac{\lambda}{l} \mathbb{E}_\Sigma \left[\left(\sum_{j=1}^m |w_j|^q \right)^{1/q} \right] \\
&\leq \frac{\lambda}{l} \left[\mathbb{E}_\Sigma \sum_{j=1}^m |w_j|^q \right]^{1/q} \\
&= \frac{\lambda}{l} \left[\sum_{j=1}^m \mathbb{E}_\Sigma |w_j|^q \right]^{1/q} \\
&\leq \frac{\lambda}{l} \left[\sum_{j=1}^m \mathbb{E}_\Sigma \left(\text{trace}(\Sigma^T \vec{\mathbf{K}}_j \Sigma) \right)^{q/2} \right]^{1/q} \\
&\leq \frac{\lambda}{l} \left[\sum_{j=1}^m \left(\eta_0 q \text{trace}(\vec{\mathbf{K}}_j) \right)^{q/2} \right]^{1/q} \\
&\leq \frac{\lambda \sqrt{\eta_0 q}}{l} \left[\sum_{j=1}^m \left(\text{trace}(\vec{\mathbf{K}}_j) \right)^{q/2} \right]^{1/q},
\end{aligned}$$

where $\eta_0 = 23/22$. Here the second inequality follows by Hölder's inequality, the fourth inequality follows by Jensen's inequality, the fifth inequality can be obtained by a similar reasoning as in the single kernel case, and the last inequality follows by straightforward extension of Lemma 1 in [10]. Let $\mathbf{u} = [\sqrt{\text{trace}(\vec{\mathbf{K}}_1)}, \dots, \sqrt{\text{trace}(\vec{\mathbf{K}}_m)}]$. We obtain, for any $q > 0$,

$$\hat{R}_l(\mathcal{H}_\lambda^1) \leq \frac{\lambda \sqrt{\eta_0 q}}{l} \|\mathbf{u}\|_q$$

Now for the special case of separable kernels we have $\|\mathbf{u}\|_q \leq \left(\sum_{i=1}^m \sqrt{l \kappa \tau^q} \right)^{1/q}$ and thus

$$\hat{R}_l(\mathcal{H}_\lambda^1) \leq \frac{\lambda \sqrt{\eta_0 q \kappa \tau}}{\sqrt{l}} m^{1/q}$$

Noting that for $m > 1$ the function $q \rightarrow q m^{2/q}$ reaches its minimum for $q = 2 \log(m)$, we get

$$\hat{R}_l(\mathcal{H}_\lambda^1) \leq \frac{\lambda \sqrt{\eta_0 2e \log(m) \kappa \tau}}{\sqrt{l}}$$

□